

Job experience

- since 2015 – [2GIS](#). Leading programmer.
 - Maintained a CDN delivering data to 20 millions of users. Converted CDN-controlling CLI for humans to a well-structured RESTful API for robots. Spent a fair amount of time improving Python code quality and performance.
 - Built fault-tolerant cross datacenter pipeline based on Kafka. It replaced the old single datacenter ZeroMQ-based pipeline. Made the transition seamless for consumers.
 - Designed JSON-based RESTful API to replace Apache Thrift API for a search backend of the online version of 2GIS. The backend required a good performance. Optimized and metaprogrammed in C++ for it.
 - Brought a new life to a dying Python-based geopositioning service. It was slow, hard to scale and fault intolerant. Rewrote and stabilized it using asyncio. As load increased with years, rewrote it in Go. Now it handles heavy load without any special requirements to a hardware.
 - Did a wide range of DevOps/SRE work. Knows containers (Docker, LXD), orchestration (Ansible), monitoring (Grafana, Zabbix), CI (Jenkins) and logs (syslog, Elastic, Kibana).
 - Wrote many documents. It's design decisions, analytics, API and service overviews, tutorials, SLAs and postmortems.
 - Created and executed several long-term team roadmaps. Decomposed, estimated and prioritized hundreds of tasks. Actively participated in sprint planning and process improvement.
 - Coordinated cross-team feature development. Has been preparing releases with dozens of branches for years.
- since 2017 – [Novosibirsk State University](#). Assistant professor.
 - Taught courses on operating systems (system calls), parallel programming (pthreads) and computing history.

Education

- 2009 – 2015. [Novosibirsk State University](#). Master's degree in Information Technology.

Languages

- *Python*. Has been using it almost every day for the past 5 years. Wrote a thousand of tests using pytest. Rewrote half of the project to satisfy mypy, increase maintainability and find old hidden bugs. Values well-designed libraries like aiohttp, SQLAlchemy, Requests and Keras.
- *C++*. Started the career using this language (C++14) but rarely sees it even in reviews nowadays. Skills are rusty but ready to be used again. There's a [code example](#).
- *Go*. Used it as a DSL for network applications. Some services were successful, some were not. Rewrote the old Python service in it, it was very satisfying.
- *Scala*. Used it not because complex languages are alluring but because it allows to build high quality Kafka-based services. It's hard to achieve the same quality in other languages. Knows some facts about functional programming. Those facts help to write better code in any language.

Interests

- *Learning*. Lifelong learner. There's a [bookshelf](#).
- *Strategy*. Likes to come up with a plan and analyze it down to small, almost invisible details. Usually it works out well because final solution is as simple as possible.
- *GNU/Linux*. Happy Gentoo/Ubuntu user for a decade. Knows enough about UNIX-like OSs to teach it at the university.
- *Distributed systems*. Defended a thesis on this topic, it's an old passion. Appreciates the motto "Reliability, Maintainability, Scalability!". Can explain what's happening in tcpdump + wireshark, Zookeeper and WALs.
- *Documentation*. Tries hard to explain complex things in a simple manner. Did deep research on this topic.
- *UX*. Prefers backend just because doesn't like programming GUIs. But cares deeply about UX.
- *Databases*. There's a list he's happy to discuss and use: Redis, Tarantool, PostgreSQL and ClickHouse.
- *Communication*. Solved a lot of "technical" problems by a productive conversation. Knows how to find real problems in generic complaints. Knows common cognitive biases. Strives to be a great listener, not a speaker.